

Assignment-A5 (Sliding Window) – Output

Code

```
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
class SlidingWindow {
private:
    int window_size;
    int total_frames;
    vector<int> sent_frames; // queue of sent frames
    int next_frame_to_send; // index of the next frame to send
    int next_frame_to_receive; // index of the next frame to receive

public:
    SlidingWindow(int window_size, int total_frames) {
        this->window_size = window_size;
        this->total_frames = total_frames;
        this->next_frame_to_send = 0;
        this->next_frame_to_receive = 0;
    }
    // send a frame
    void send_frame() {
        if (sent_frames.size() < window_size && next_frame_to_send <
total_frames) {
            sent_frames.push_back(next_frame_to_send);
            next_frame_to_send++;
            cout<<"Sent frame "<<sent_frames.back()<<endl;
        }
        else {
            cout<<"Window is full. Waiting before sending next frames."<<endl;
        }
    }
    // receive a frame
    void receive_frame(int frame_number) {
        if (frame_number == next_frame_to_receive) {
            next_frame_to_receive++;
            cout<<"Received frame "<<frame_number<<endl;
        }
    }
}
```

```

    else {
        cout<<"Received frame "<<frame_number<<", but expected frame
"<<next_frame_to_receive<<endl;
    }
    // remove acknowledged frames from the sent_frames queue
    while (!sent_frames.empty() &&
sent_frames.front()<=next_frame_to_receive-1) {
        sent_frames.erase(sent_frames.begin());
    }
}
bool all_frames_sent() {
    return next_frame_to_send == total_frames;
}
bool all_frames_received() {
    return next_frame_to_receive == total_frames;
}
};

int main() {
    int window_size, total_frames;
    cout << "Enter the window size:\t";
    cin >> window_size;
    cout << "Enter the total number of frames:\t";
    cin >> total_frames;

    SlidingWindow window(window_size, total_frames);

    while (!window.all_frames_sent() || !window.all_frames_received()) {
        window.send_frame();
        // simulate receiving frames
        for (int i=0; i<rand()%3; i++) {
            int frame_number = rand() % total_frames;
            window.receive_frame(frame_number);
        }
    }
    cout<<"All frames sent and received successfully!"<<endl;

    return 0;
}

```

Output

```
[overnion - Codes (/run/media/overnion/persistence/Fil
$ g++ Code-A5\ \(\Sliding\ Window\).cpp && ./a.out
Enter the window size: 3
Enter the total number of frames:      5
Sent frame 0
Received frame 1, but expected frame 0
Sent frame 1
Received frame 3, but expected frame 0
Sent frame 2
Received frame 2, but expected frame 0
Window is full. Waiting before sending next frames.
Received frame 2, but expected frame 0
Window is full. Waiting before sending next frames.
Received frame 4, but expected frame 0
Received frame 1, but expected frame 0
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 1, but expected frame 0
Received frame 3, but expected frame 0
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 0
Received frame 3, but expected frame 1
Sent frame 3
Received frame 4, but expected frame 1
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 2, but expected frame 1
Window is full. Waiting before sending next frames.
Received frame 1
Sent frame 4
Received frame 4, but expected frame 2
Window is full. Waiting before sending next frames.
Received frame 4, but expected frame 2
Received frame 0, but expected frame 2
Window is full. Waiting before sending next frames.
Received frame 1, but expected frame 2
Received frame 1, but expected frame 2
Window is full. Waiting before sending next frames.
Received frame 0, but expected frame 2
Window is full. Waiting before sending next frames.
Received frame 0, but expected frame 2
Window is full. Waiting before sending next frames.
Received frame 2
Received frame 0, but expected frame 3
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 2, but expected frame 3
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 2, but expected frame 3
Received frame 4, but expected frame 3
Window is full. Waiting before sending next frames.
Received frame 2, but expected frame 3
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 3
Window is full. Waiting before sending next frames.
Window is full. Waiting before sending next frames.
Received frame 4
All frames sent and received successfully!
```